Technical Perspective: ϵKTELO: A Framework for Defining Differentially-Private Computations

Graham Cormode
University of Warwick, UK
G.Cormode@warwick.ac.uk

When was the last time that you wrote code to implement a join algorithm? Chances are, it was during an undergraduate database class – if at all. The wide availability of database management systems in all their manifestations (admitting a wide definition, to encompass performing look-ups in a spreadsheet) mean that we do not have to (re)implement common operations over and over again. This brings many advantages. We benefit from time savings, both in development time, and also in execution time: we can expect that optimized professional code will outperform our ad-hoc efforts. Moreover, we expect such code to be robust, and less prone to crashing on unexpected inputs. It should produce results that can be relied on to be correct, and handle errors gracefully.

Such "systematization" is a core methodology in computer science. Whenever we identify new areas where computation is needed, there is a sequence of steps that can be followed. First, we develop algorithms for special cases or particular operations. Over time, these move from proof-of-concept code into more reliable libraries and toolkits. From these, we abstract new collections of operations that together can be combined to address instances that might arise. Describing the sequence of steps to perform might initially be done via simple scripting or calls out from an existing high-level language, but over time may instead be expressed via a special purpose (and oft-times declarative) language, or through a graphical user interface. Eventually, we have a stand-alone system to describe tasks, which can be deployed by users who might otherwise lack the ability to code up the routines themselves.

Viewed through this lens, we can see many cases of systems emerging in computer science. The (relational) database management system is perhaps our default example. High-level languages themselves have also followed this path. Currently, machine learning tools are part way through this evolution: machine learning algorithms and libraries have been around for a while, but we are yet to achieve user-friendly systems that allow non-expert users to quickly and easily define complex machine learning pipelines.

The following paper by Zhang *et al.* talks in terms of a framework for a class of privacy-preserving computations over data. The artifact, ϵKTELO, represents an important step on the pathway to providing systems for such computations. It is not the first system in this domain. Frameworks such as PINQ [2], which extends the (non-private) LINQ framework, and Featherweight PINQ [1] are acknowledged as direct antecedents. ϵKTELO extends these by providing a different selection of primitives at higher levels of abstraction.

Computing under guarantees of privacy shares many similarities with secure computation. In particular, the mantra "Don't roll your own crypto" can equally well be transcribed as "Don't roll your own privacy". Subtle (and not so subtle) errors in defining and combining algorithms to protect the privacy of individuals motivate us to further automate and systematize the handling of private data. ϵKTELO assists by not only providing a broad set of tools for the most common operations on private data, but also by simplifying the analysis of the privacy properties of the resulting composition (captured by the sometimes inscrutable privacy parameter ϵ alluded to in the name). It more clearly separates the private from the public domain, and directs attention to steps which move information across this divide. The value of the framework is demonstrated not only by the ease with which a range of algorithms can be expressed, but also in the way it exposes new variants that can lead to improved utility.

By no means should we expect ϵKTELO to be the last word on this topic. There are a number of limitations that we can hope future work to overcome. Chief among these is the restriction to input that is modeled as a single table. Compare this to the DBMS, where we can SELECT, PROJECT and JOIN to our heart's content, all the while remaining fully within the world of relations. The ability to represent and compute by linking over different tables, and model networks of interactions, represents a pressing open problem for privacy computation that needs to be solved before systems can generalize to this case.

1. REFERENCES

[1] Hamid Ebadi and David Sands. Featherweight PINQ. *Journal of Privacy and Confidentiality*, 7(2):159–184, 2017.

[2] Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD Conference*, 2009.